Proc. of the 8th International Conference on Engineering and Emerging Technologies (ICEET)
27-28 October 2022, Kuala Lumpur, Malaysia

Blockchain-based Approach to thwart Replay Attacks targeting Remote Keyless Entry Systems

Issa Abdoul Razac Djinko
Department of Computer Science
and information Technology
University of the District of Columbia
Washington, DC, USA
issaabdoulrazac.djin@udc.edu

Thabet Kacem
Department of Computer Science
and information Technology
University of the District of Columbia
Washington, DC, USA
thabet.kacem@udc.edu

Anteneh Girma
Department of Computer Science
and information Technology
University of the District of Columbia
Washington, DC, USA
Anteneh.Girma@udc.edu

***Abstract*—Remote Keyless Entry (RKE) is a revolutionary technology that allows drivers to gain access to their vehicles using a wireless key fob operating on short-range radio waves. This technology offers numerous advantages for drivers, but it is still vulnerable to serious security threats which target the interactions between the fob and car, which may eventually result in car theft. In this paper, we propose a blockchain-based approach to thwart replay attacks targeting cars equipped with RKE systems. We consider the key fob and the car as two separate users of a private blockchain in which a miner authenticates the key fob with the corresponding car to grant or deny access using a smart contract. We validate our findings by a performance evaluation of the time taken by the miner to validate the access to the vehicle.**

***Index Terms*—RKE, Blockchain, Replay Attacks, Elliptic Curve Digital Signature**

I. Introduction

In 2013, it was estimated [14] that the global market for connected vehicles would reach 131.9 Billion by 2019. Today, vehicles are no longer expected to just run, they also have wireless connectivity and even a driver assistance systems to allow them to drive with little to no human assistance.

Conversely, technology is always changing and improving as time goes on as Moore's law states that our computational power doubles every two years [1]. This fact also implies that attackers are also getting smarter and more capable. Connected vehicles have always been targeted because of their capability to be remotely connected. In particular, the RKE systems in cars can be hacked and impersonated by bad actors leveraging weak encryption methods to copy access codes from the key fob and potentially steal the car and/or its content.

There are many ways that attackers can use to take control of connected Vehicles [7], [8], [9]. One of the most common attacks is the replay attack that is a process of copying responses given by the key fob and transmitting them later to the receiving end of the RF interaction in order to gain access.

On the other hand, blockchain is a promising technology that allows decentralized and secure access of data in a distributed ledger using cryptography. Actually, it has been very popular with crypto-currencies such as Bitcoin and Ethereum. It has also been used to secure peer-to-peer communications in several domains [5] and have the potential to be applied to RKE systems.

In this paper, we propose a blockchain-based approach to thwart replay attacks. The key fob and the car are considered as two users in a private blockchain, in addition to a miner that validates the access to the car using a smart contract. We validate our findings by measuring the block mining time that refers to the time taken by the miner to mine each block thus validating access to the vehicle.

This rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 shows the background of RKE systems. Section 4 investigates threat model of RKE. Section 5 explains our approach in detail. Finally, section 6 presents our simulation and the results.

II. Related Work

Steger et al. [13] proposed a distributed solution to automotive security and privacy. The paper integrated the vehicle into an IoT environment to allow every device to transact securely. Transactions were verified by the overlay block managers that match the public keys and messages with the private keys. The approach is different from ours in the sense that we do not consider other IoT devices.

KiranRaj et al. [6] analyzed the security of RKE by launching attacks on six car models using several configuration system parameters. In our case, we are using blockchain to counter replay attacks.

PcCoe [7] proposed the use of symmetric cryptography algorithms such as AUT64 cipher and others to protect RKE systems. However, vulnerabilities are still apparent despite the use of all those encryption algorithms.

Greene [8] proposed a timestamped-based method to secure RKE. The paper reviewed the potential attacks on the RKE and concluded that the median used for communication between the key fob and car is the point of vulnerability to RKE.

III. RKE Background

The first RKE system was developed and implemented by Renauld [2] in 1982. It goes without saying that before 1982,

This work was funded by NSF award 2011689 and NSA grant H98230-20-1-0289

the only choice was a physical key to gain access and control of one's car [7].

Progress have continued since in terms of car keys. The next step was a physical key with an immobilizer. This was a big step up in this area as the immobilizer was preventing the key from being replicated [7].

Afterwards an old fashion RKE system emerged to allow the user to interact with, lock or unlock the car. This key still has a physical key that needs to be inserted into the ignition and turned to start the car [7]. Indeed, the RKE has a couple of buttons that send radio frequency signals to control the access to the car.

The passive RKE system does more than just locking and unlocking cars with the push of a button. Most of them still have this feature but in fact, the car interacts with the key fob to automatically unlock the door when the user comes near and touches the knob. Moreover, it lets the user press a start and stop button to start the car without the insertion of a key [7]. As long as the key fob is inside the car, it can be started.

The design of the RKE system is shown in Figure 1. Both the key fob and the receiver have the same components: an antenna, a CPU, a Radio Frequency (RF) transmitter, a power supply, a command module, and push-button switches for the fob.

When the button on the Fob is pressed, a process is started thus waking the CPU [16]. A stream of 64 to 128 bits is generated and modulated, then sent through the transmitter and antenna.

The receiver gets the data stream via the signal sent by the fob, the data is demodulated and sent to the CPU for further decoding. If the data is verified, a command is sent to the vehicle's lock system [16].

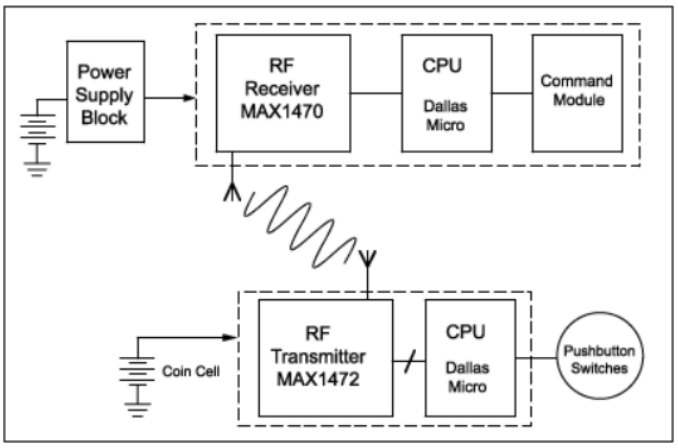

Fig. 1. Block diagram for remote keyless entry (RKE) systems [16]

IV. RKE Threat Model

A. Vulnerabilities

RKE has several vulnerabilities, stamping from its use of radio frequency devices. The term CIA (Confidentiality, Integrity, and Availability) refers to the importance of keeping information at rest or in transit safe, unchanged, and available to intended users [10]. First, with regards to availability, the signal can be jammed, thus jeopardizing the interaction. Second, with regards to integrity, the signal can be repeated thus impersonating the key fob. Finally, replay attacks seem to thwart both the confidentiality and integrity by replaying the signal from the key fob and being able to change it as well.

Moreover, RKE technology can also be the victim of many other different attacks. Some are less sophisticated than others. Consequently, there needs to be a better method to counter all of these potential attacks. To sum up, the RKE vulnerabilities can be categorized into three classes depending on the RKE component, as follows:

1) Key Fob: The key fob can be stolen because it is just like a traditional key. If it is misplaced and a bad actor finds it or steals it, they can consequently steal the car and never be found again.

2) Wireless Connection: The main premise of RKE is that the car can remotely be interacted with. This makes it vulnerable to potential remote hacking. The signal generated to allow for interaction between the key fob and the car can be the exploit that bad actors can take advantage of.

3) Human Factor: In the old days, one needed to insert a physical key into the car's door knob to lock or unlock it and to start and/or stop the engine. In an modern RKE systems, the key fob is used to remotely start the car and when the driver arrives at a destination, if he forgets to stop the engine, the car equipped with an RKE system does not automatically stop the engine. Consequently, mistakes like those can provide an opportunity for thieves to steal the contents of the car.

B. Attacks

1) ***Relay Attack****:* This attack, as shown in figure 2, consists of one entity using a device much like a repeater to relay the signal and give the impression of the key fob being very close. It tricks the car into letting an intruder gain access [7]. Since RF is the medium for the communication between the key fob and the car, using an appropriate device, such an attack is feasible to accomplish. Furthermore, it can prove very difficult to counter as it does not leave any trace and may make the car owner think he/she accessed the car.

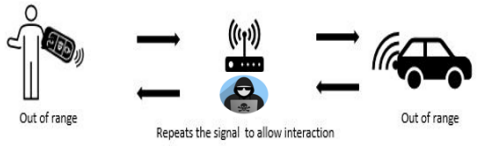

Fig. 2. Relay Attack

2) ***Replay Attack****:* This type of attack occurs when the data in transit between the fob and the car is captured then replayed by attackers as shown in figure 3. The attacker intercepts the packets and reads them making the communication either delayed and/or repeated [7]. The bad actor can thus edit the data or use it at a later time and impersonate the victim.

Nowadays the key fob receives challenges from the car then responds with the corresponding data to unlock the vehicle [9]. At the end of the seemingly normal interaction between the key fob and the car, using a sophisticated device to capture the data, the attacker can get the corresponding response. Later, the hacker can interact with the car and gain access by re-transmitting the stolen data.

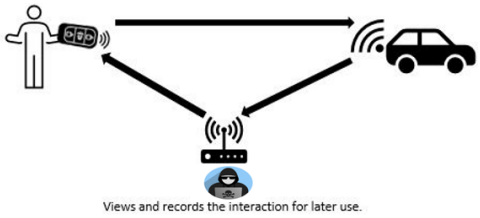

Fig. 3. Replay Attack

3) ***Radio Jamming Attack****:* Since the RKE systems uses radio frequency (RF) to interact, radio jamming attacks use a RF device to interfere with the communication [7] as shown in figure 4. The basic premise here is to prevent the vehicle's owner from locking the car by pressing the lock button on the fob. The weakness of this attack is that the perpetrator will not be able to steal the car but will only be able to come in possession of objects in the victim's car.

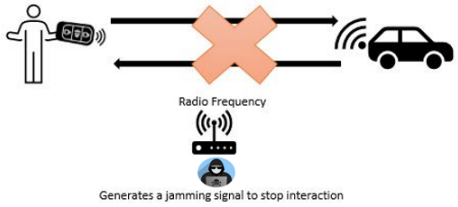

Fig. 4. Radio jamming attack

C. Blockchain Background

At a basic level, a Blockchain is a decentralized ledger system that records transactions securely. A record, also called a block, is comprised of many transactions. Blocks are added to the blockchain by nodes on the network called miners. Every node on the network is connected in the peer-to-peer manner. What makes blockchain revolutionary is that every block is connected to the previous block in a manner that makes a single change affect every previous block [3]. This is guaranteed by a hash. There are two types of blockchains: public/permissionless blockchain and private/permissioned blockchain.

Anyone can participate in a permissionless blockchain as it is available to the public, where every block is published and available for everyone to see. As mentioned before, miners add blocks to the blockchain and everyone monitors the transactions in the blocks. There is no centralized owner of this type of the blockchain.

Permissioned blockchain is the opposite of public blockchain, as there are a limited number of users. Also, everyone has to get approval before joining and the information is not available for all to see. There is a great deal of privacy kept in this type of blockchain. In fact, the chosen node in the network has the right to approve and write blocks in the blockchain, therefore making the private blockchain partially decentralized [15].

V. Proposed Approach

A. Approach Rationale

Blockchain is a decentralized ledger System that uses a strong encryption algorithm to render the data possible to be decrypted. It furthermore guarantees the integrity and availability of records [10]. That was our main motivation of applying that level of transparency and security to secure the RKE system. The implementation of blockchains in RKE technology is a game-changer in the automobile sector.

We provide in the next paragraphs key points portraying the rationale behind our approach.

First, instead of a public or permissionless blockchain, we have chosen to adopt a private/permissioned blockchain. The reason for that is quite simple. In public blockchains, every transaction is visible to everyone making transparency a crucial part of the blockchain. In permissioned blockchain, a selected group of nodes are chosen to be miners and not everyone can be a user [11]. This adds an extra layer of security to private blockchains.

Second, we propose the addition of a smart contract to manage the access to the car. Smart contracts are programs included in the blockchain to allow automatic actions based on predetermined conditions [11]. The use of this tool has greatly allowed transactions between businesses to be automated and recorded in the blockchain. Therefore, it acts as an impartial non-third party to guarantee well completion of tasks in complete transparency.

Third, blockchain uses an asymmetric encryption by which each user is assigned a private and a public key that are interlinked. We used the Elliptic Curve Digital Signature Algorithm (ECDSA) as a digital signature scheme [11]. This is used to accomplish three extremely important tasks. The first is to generate the private and public interlinked keys. The second task is to sign the message with the private key to ensure it really comes from the sender's public key. The third task verifies the whole transaction by taking as input the signature, message, and receiver's public key then outputs true or false based on the veracity of the inputted items.

Finally, each transaction is sent to the transaction pool waiting to be added to the blockchain. It should be mentioned that since we opted for a permissioned blockchain, the appropriate consensus algorithm will be proof of authority, thus verifying transactions faster than in a permissionless blockchain [11]. When the transaction is verified and cleared, an "OK" message will be sent to the vehicle, allowing it to grant access. Later, all cleared transactions will be added to the private blockchain in form of blocks.

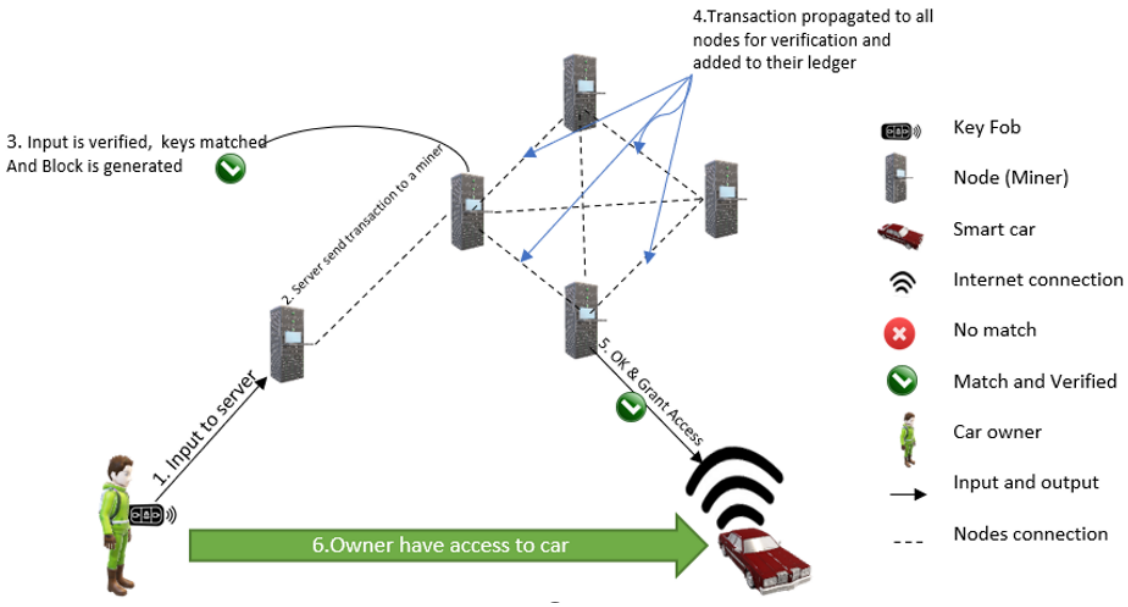

Fig. 5. System Model

B. System Model

Our system model is described in figure 5 where the process of gaining access to the vehicle by a transaction is explained. An authentication stage takes place before an authorization is granted. At this stage, we assume that the key fob initiates the transaction because it is the user that needs to authenticate itself.

First, the key fob uses the vehicle's public key to encrypt the the message that consists of the command to be executed. Then it signs the message with its private key. The input of the transaction is the public key of the car, the signature of the message using its private key, and the message itself. The transaction is then sent to the server.

Second, the transaction is received by the server saying " hey this user by this public key would like to interact with this other user, also by the car's public key". The server then chooses which miner should process the transaction [12].

Third, when the miner receives the transaction, it authenticates the public key with its private key by looking at the signature. It also compares the message with the one in the signature. If everything clears, it will generate a 32-byte hash of the transaction with the hash for the previous block and sends it to the rest of the nodes on the peer-to-peer blockchain network for more verification. Finally, when all nodes agree with the transaction, an "OK" message is sent to the car that grants access to the key fob.

C. Smart Contract

Contrary to public blockchain where every authorized user can see the mined blocks, in our private blockchain, the data should only be available to the user who will have to verify its credentials before getting access to the data on the blockchain. The vehicle access is governed by a smart contract described in Algorithm 1.

In lines 2-11, if the user needs to view block data to access the vehicle or to decide whether to grant access or not, it needs to verify its credentials.

In lines 12-17, if for any reason an administrator needs to see a specific user's data, it will need to send a request to that user who will grand access by entering their own credentials. In case the user cannot grant access or will not grand access, upper management may override the system and grant permission.

In lines 18-24, if bad actors attempt to gain access, they will enter their credentials and when it is not verified, the access is denied.

VI. Simulation

A. Prerequisites

First, we installed the right version of Geth , also known as go-Ethereum. This turns a computer into a node on the Ethereum network that can compile blocks and add them to the blockchain. The software can be found on the Ethereum website. Five different versions of the software can be found: Android, iOS, macOS, Windows, and Linux. For this paper, we used a windows laptop and we downloaded Geth version 1.7.3.

The second condition is to install atom. There are many text editors but we chose to atom to edit and add instructions to some of our directives.

The third condition is to create a nested directory whereby all the necessary files are saved. In this study, we chose to

Algorithm 1: Smart Contract for Vehicle Access

```
Inputs: A=user, Y=admin, X=Bad actor;
while (access is granted) do
    if A requests access then
        Enter username and password;
        if credentials are valid then
            Grant access
        end
        else
            Start over
        end
    end
    if Y request access then
        Send request to user
    end
    else
        Override with management credentials
    end
    if X request access then
        Enter username and password;
        if credentials are not valid then
            Grant denied
        end
    end
end
```

create a directory called ”university” and within that directory, we created another one called ”privateB” which stands for private blockchain.

B. Creating the Node

When we installed geth, we also installed puppeth which is part of the package. After creating the different directories, to create the node, the puppteh command needs to be executed in the privateB directory using Windows Powershell. Puppeth first gives the option to input a name for the node. Upon adding the chosen name, we chose option 2 which is to configure a new genesis for our private blockchain.

The next step is to choose a consensus algorithm. For this study, we decided, as mentioned earlier, the Proof of Authority as most private blockchains operate with this algorithm. [17]. Figure 6 shows the system asking which consensus engine will be used and we chose the second option.

```
What would you like to do? (default = stats)
 1. Show network stats
 2. Configure new genesis
 3. Track new remote server
 4. Deploy network components
> 2

Which consensus engine to use? (default = clique)
 1. Ethash - proof-of-work
 2. Clique - proof-of-authority
> 2
```

Fig. 6. Proof of Authority

Next, the following question needed to solved: how many seconds should the block take to be validated? The default option is 15 seconds. We chose one second because in this study we need the blocks to be created as fast as possible and all transactions to be processed and added to the blockchain as the key fob approached the car.

After creating and exporting the genesis block, which is shown in figure 7, we needed to create three directories: the first was is ”startnode.cmd” to hold the different instructions to be able to start our node, the second named ”keystone” to hold the different accounts that are needed for transactions and the last one called ”geth” to act as storage. When these steps are completed, the node can be fired up to start mining blocks.

Fig. 7. Genesis Block

C. Starting the Node

After meeting the prerequisites, we started the node by going to the directory where all the files are stored in the Windows PowerShell and entering the command: *startnode.cmd*.

D. Transactions

In our approach section, we mentioned that we were considering the key fob and the car receiver as two different accounts that run transactions between them. For the proof of concept, we created two accounts, in an ideal world this should be done by the car manufacturer using the command *geth –datadir . account new*. Then, we created a passphrase to enforce security.

The next step is to submit a transaction from the first account that represents the key fob to the second account which is the car in this case using sendTransation command.

E. Results of the Simulation

It is widely known that the average time it takes for transactions to be confirmed and placed in a block to be mine

in a public blockchain is around 10 minutes [3]. This is not the case for private blockchains and this was the sole reason why permissioned blockchains fit very well with our approach. Many actors impact the latency of the transactions. In a private blockchain environment that uses PoA with a set time for blocks to be mined, the latency time is greatly diminished. Figure 8 shows the that it took one second for the node to mine a block. Figure 9 shows two different nodes processing and adding blocks in a private blockchain environment. The results show an average of 0.016 second of latency per transaction.

Fig. 8. One second of mining time Block

P	# Transactions / block		Latency (sec.)	
	M1	M2	M1	M2
2	145	127	0.016	0.019
4	309	270	0.015	0.018
6	435	368	0.016	0.020
8	589	559	0.015	0.016
10	785	728	0.015	0.016
12	**1010**	**950**	**0.013**	**0.014**
14	997	918	0.016	0.018
16	1152	1048	0.016	0.018
18	1258	1172	0.016	0.018
20	1619	1592	0.013	0.013
22	1641	1376	0.015	0.019
24	2022	2009	0.023	0.031
26	2023	2022	0.025	0.042

Fig. 9. PoA results for 2 nodes [17]

VII. Conclusion

RKE is a promising technology that facilitate access to vehicles but it has several vulnerabilities that pose great risk of attacks against it. We proposed a blockchain-based approach to thwart replay attacks on the RKE system. There may be many different ways to use blockchain to that end. However, we chose to use a private blockchain where we considered the key fob and the car as two different blockchain users. This allowed both users to submit transactions to the blockchain to be added to the different blocks. Our simulation and corresponding results show promising results to validate transactions thus, making the car grant access instantly and securely.

In future work, we will physically separate the two users of the permissioned blockchain to better represent a real-life scenario. There will be more transactions to be run to further study the time it takes for transactions to be cleared and added to the blockchain. A smart contract will also be run simultaneously with the blockchain to guarantee that the transactions for each car can be secured and only visible to the owner or only when the permission is granted. Furthermore, to capture the sent and received data by the car, we will introduce a device capable of doing just that. The data will then be analyzed and checked to make sure that it is useless to potential bad actors that may attempt to commit replay attacks.

References

[1] Schaller, R. R. (1997). Moore's law: past, present and future. IEEE spectrum, 34(6), 52-59.

[2] Craig Smith. The Car Hacker's Handbook: A Guide for the Penetration Tester. No Starch Press, 2016.

[3] Tasatanattakool, P., & Techapanupreeda, C. (2018, January). Blockchain: Challenges and applications. In 2018 International Conference on Information Networking (ICOIN) (pp. 473-475). IEEE.

[4] Ibrahim, O. A., Hussain, A. M., Oligeri, G., & Di Pietro, R. Key Is In The Air: Hacking Remote Keyless Entry Systems.

[5] Dorri, A., Kanhere, S. S., Jurdak, R., & Gauravaram, P. (2017, March). Blockchain for IoT security and privacy: The case study of a smart home. In 2017 IEEE international conference on pervasive computing and communications workshops (PerCom workshops) (pp. 618-623). IEEE.

[6] KiranRaj, K. G., Khot, S., Choudhary, A., & Singh, R. (2019). ANALYSING REMOTE KEYLESS ENTITY SYSTEMS. IJRAR-International Journal of Research and Analytical Reviews (IJRAR), 6(2), 136-138.

[7] PCCOE, P. P. P. On Vehicular Security for RKE and Cryptographic Algorithms: A Survey.

[8] Greene, K., Rodgers, D., Dykhuizen, H., McNeil, K., Niyaz, Q., & Al Shamaileh, K. (2020, January). Timestamp-based defense mechanism against replay attack in remote keyless entry systems. In 2020 IEEE International Conference on Consumer Electronics (ICCE) (pp. 1-4). IEEE.

[9] Wouters, L., Marin, E., Ashur, T., Gierlichs, B., & Preneel, B. (2019). Fast, furious and insecure: Passive keyless entry and start systems in modern supercars. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2019(3), 66-85.

[10] Zhou, M., Zhang, R., Xie, W., Qian, W., & Zhou, A. Security and Privacy in Cloud Computing: A Survey.

[11] Liu, M., Wu, K., & Xu, J. J. (2019). How Will Blockchain Technology Impact Auditing and Accounting: Permissionless versus Permissioned Blockchain.

[12] Gupta, S., & Sadoghi, M. (2021). Blockchain transaction processing. arXiv preprint arXiv:2107.11592.

[13] Dorri, A., Steger, M., Kanhere, S. S., & Jurdak, R. BlockChain: A distributed solution to automotive security and privacy.

[14] Lu, N., Cheng, N., Zhang, N., Shen, X., & Mark, J. W. (2014). Connected Vehicles: Solutions and Challenges. IEEE INTERNET OF THINGS JOURNAL, 1(4), 289.

[15] Liu, M., Wu, K., & Xu, J. J. (2019). How Will Blockchain Technology

[16] Impact Auditing and Accounting: Permissionless versus Permissioned Blockchain. False. (n.d.). Designing remote keyless entry (RKE) syst: Maxim integrated. Designing Remote Keyless Entry (RKE) Syst — Maxim Integrated. Retrieved June 9, 2022, from https://www.maximintegrated.com/en/design/technical-documents/app-notes/1/1773.html

[17] Leal, F., Chis, A. E., & González–Vélez, H. (2020). Performance evaluation of private ethereum networks. SN Computer Science, 1(5), 1-17.